

# 用模拟退火思想的粒子群算法 实现图像分割

张捍东, 廖天红, 岑豫皖

(安徽工业大学 电气信息学院, 安徽 马鞍山 243002)

**摘 要:**采用了一种模拟退火思想的粒子群算法与最大类间方差法相结合的快速阈值分割法对图像进行分割。用粒子群优化算法来搜索阈值向量, 每个粒子代表一个可行的阈值向量, 通过粒子间的协作来获得最优阈值。为了提高收敛速度, 把模拟退火的思想应用在粒子群算法中, 最后仿真结论表明, 该方法在继承标准粒子群算法原理简单、易于实现、协同搜索等优点的同时, 还避免了标准粒子群算法的收敛速度慢问题, 有更强的寻优能力, 得到理想的结果的同时计算量大大减少。权衡分割精度和计算效率两个方面, 文中方法不失为一种实用有效的图像分割算法。

**关键词:**模拟退火; 粒子群算法; 最大类间方差法; 阈值分割

中图分类号: TP301

文献标识码: A

文章编号: 1673-629X(2010)05-0083-05

## Image Segmentation Through Particle Swarm Optimization Based on Simulated Annealing

ZHANG Han-dong, LIAO Tian-hong, CEN Yu-wan

(Dept. of Electrical Engineering & Information, Anhui University of Technology, Maanshan 243002, China)

**Abstract:** A rapid Otsu's method based on simulated annealing particle swarm optimization algorithm was used for image segmentation. Particle Swarm Optimization (PSO) algorithm was used to search threshold vectors. Each particle represents a feasible threshold vector. Thus, the optimal threshold could be acquired by the cooperation of particle swarm. To get better convergence, simulated annealing idea was applied in PSO algorithm. Simulation experiment results demonstrated that this method retains the uncomplicated principle, simple-operation and coevolutionary search of standard PSO, and also solves the slow convergence problem of standard PSO, could acquire ideal results with less computation. The algorithm is practical and effective in image segmentation considering segmentation accuracy and computing efficiency.

**Key words:** simulated annealing; particle swarm optimization; Otsu; threshold segmentation

## 0 引言

图像分割是一种重要的图像技术, 它不仅受到人们广泛的重视, 其研究在不断深入, 而且也在实际中得到了大量的应用。图像分割是应用一种或多种运算将图像分成一些具有类似特征(如颜色、纹理、密度等)的区域, 主要方法有阈值法和区域方法两大类。前者是利用灰度频率信息进行分割, 而后者利用局部空间信息进行分割。

其中阈值方法由于其简单性和有效性从而得到广

泛的使用<sup>[1]</sup>。阈值分割方法很多, 其中应用较多的有最大类间方差法(也称 Otsu 法)<sup>[2]</sup>、最小误差法、熵方法等。Otsu 法于 1978 年由日本学者大津提出。Otsu 法因其计算简单、分割效果好一直受到人们的广泛关注。但是 Otsu 方法和其它的阈值分割方法都存在着计算量大、不适合实时处理的缺点。近年来为了简化计算, 缩短寻优过程, 将遗传算法等寻优算法应用于图像分割, 成为图像领域的研究热门。不过国内将 PSO 优化算法用于图像分割领域的相关成果却并不多<sup>[3,4]</sup>。

正是基于缩短寻优过程的考虑, 借鉴文献中把模拟退火方法引入量子空间模型粒子群优化算法的思路<sup>[5]</sup>, 文中采用一种模拟退火思想的粒子群算法(Simulated Annealing Particle Swarm Optimizer, SAPSO)结合最大类间方差方法进行图像分割。该方法既克服了最

收稿日期: 2009-09-29; 修回日期: 2009-12-26

基金项目: 国家自然科学基金项目(50407017); 安徽省自然科学基金重点项目(2006KJ019A, 2007KJ052A)

作者简介: 张捍东(1963-), 男, 安徽桐城人, 教授, 博士, 研究方向为智能控制理论与应用、机器人运动控制、计算机控制及相关技术等。

大类间方差法计算量大、不适合实时处理的缺点,又避免了基本粒子群算法可能存在的收敛速度慢问题,文中方法寻优能力强,与已有算法相比具有更快的运算速度,实时性更好。

## 1 基于模拟退火思想的粒子群算法

### 1.1 基本 PSO 算法

首先,粒子群算法是由 Eberhan 博士和 Kennedy 博士最先提出的全局优化进化算法<sup>[6,7]</sup>。该算法源于对鸟群捕食行为的灵感,其基本思想是通过群体中个体之间的协作和共享来寻求最优解<sup>[8~10]</sup>。

粒子群算法中,每个优化问题的解看作是搜索空间中一只鸟,称之为“粒子”。鸟被抽象为没有质量和体积的微粒(点),并将其延伸到  $N$  维空间,粒子  $i$  在  $N$  维空间里的位置表示为矢量  $X_i = (x_{i1}, x_{i2}, \dots, x_{iN})$ , 飞行速度表示为矢量  $V_i = (v_{i1}, v_{i2}, \dots, v_{iN})$ , 每个粒子都有一个由目标函数决定的适应值(FitnessValue), 并且知道自己到目前为止发现的最好位置 pbest 和现在的位置  $X_i$ , 这个可以看作是粒子自己的飞行经验。除此之外,每个粒子还知道到目前为止整个群体中所有粒子发现的最好位置 gbest(gbest 是 pbest 中的最好值)。这个可以看作是粒子同伴的经验。粒子就是通过自己的经验和同伴中最好的经验来决定下一步的运动<sup>[8]</sup>。

对于第  $k$  次迭代,PSO 中的每一个粒子是按照式(1)和式(2)进行变化的。

$$v_{id}^{k+1} = w \times v_{id}^k + c_1 \times r_1 \times (P_{id} - x_{id}^k) + c_2 \times r_2 \times (P_{gd} - x_{id}^k) \quad (1)$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \times \Delta t \quad (2)$$

在式(1)、(2)中:

$i = 1, 2, 3, \dots, M$ ,  $M$  是该群体中粒子的总数;

$v_{id}^k$  是第  $k$  代粒子  $i$  行速度矢量的第  $d$  维分量;

$c_1, c_2$  是加速常数; $w$  是惯性权重因子;

$x_{id}^k$  是  $k$  次迭代粒子  $i$  位置矢量的第  $d$  维分量;

$r_1, r_2$  是随机函数,产生  $[0, 1]$  的随机数;

$P_{id}$  是粒子  $i$  个体最好位置 pbest 的第  $d$  维分量;

$\Delta t$  是时间步长。

$P_{gd}$  是群体最好位置 gbest 的第  $d$  维分量。

粒子通过不断的学习更新,最终飞向解空间中最优解所在位置,搜索过程结束。最后输出的  $g$  是全局最优解<sup>[11,12]</sup>。在更新过程中,粒子每一维的速率被限制到  $v_{\min} - v_{\max}$  之内。

### 1.2 模拟退火思想的粒子群算法(SAPSO)

基本粒子群优化算法中,虽然粒子速度作了限制,

不会变化太大,但位置更新时未作限制,就有可能新的位置会变得很坏,引起收敛速度缓慢,所以要对更新的位置作限制。限制的思路有两种方法:一种类似于速度的限制方法,给每一维变量限制一个范围;另一种思路采用模拟退火算法思想,模拟退火算法用于优化问题的出发点是基于物理中固体物质的退火过程与一般优化算法的相似性。算法的基本思想是从一给定解开始的,从邻域中随机产生另一个解,接受准则允许目标函数在有限范围内变坏,它由一控制参数  $t$  决定,其作用类似于物理过程中的温度  $T$ ,对于控制参数  $t$  的每一取值,算法持续进行“产生新解 — 判断 — 接受或舍弃”的迭代过程,对应着固体在某一恒定温度下趋于热平衡的过程。经过大量的解变换后,可以求得给定控制参数  $t$  值时优化问题的相对最优解。然后减小控制参数  $t$  的值,重复执行上述迭代过程。当控制参数逐渐减少并趋于零时,系统也越来越趋于平衡状态,最后系统状态对应于优化问题的整体最优解,该过程也称冷却过程<sup>[8]</sup>。

方法 1: 在基本粒子群算法的基础上,在更新当前位置时,把它限制在  $x_{\min} - x_{\max}$  内。

采用模拟退火算法思想有 3 种方法改进,(在此以求目标函数的极小值为例,即适应值越小越好) 分析如下:

方法 2: 按式(2),计算新的位置,然后计算两个位置所引起的适应值的变化量  $\Delta E$ ; 若  $\Delta E \leq 0$ , 接受新值,否则若  $\exp(-\Delta E/T) > \text{rand}(0, 1)$  也接受新值。其中  $\text{rand}(0, 1)$  表示  $0 - 1$  之间的随机数,  $T$  是起始温度,  $\alpha$  为退火系数。否则就拒绝。

方法 3: 接受准则允许目标函数在有限范围内变坏,并不直接按概率取舍,而是按  $\Delta E < e$ ,  $e$  为按允许目标函数变坏的范围。

方法 SAPSO: 把方法 1 与方法 3 结合在一起,现提出一种基于模拟退火思想微粒群算法(SAPSO),既限定位置的范围,又设定允许目标函数变坏的范围  $e$ , 接受准则允许目标函数在有限范围内变坏,即若  $\Delta E < e$ , 就接受新值,否则就拒绝。把 SAPSO 方法结合最大类间方差法应用于图像分割,这充分发挥了模拟退火、PSO、最大类间方差法三种算法的优势,非线性快速且稳定地求解最大类间方差及对应的灰度阈值,在取得较好的图像分割效果的同时,显著地提高了计算速度。

## 2 基于模拟退火思想的 PSO 算法用于图像分割

### 2.1 最大类间方差分割方法

最大类间方差阈值分割法的基本思路是将直方图

在某一阈值处分割成两组,计算两组的方差信息,因为方差是灰度分布均匀性的一种度量,方差值越大,说明构成图像的两部分差别越大,当部分目标错分为背景或部分背景错分为目标都会导致两部分差别变小,因此使类间方差最大的分割意味着错分概率最小,此时的阈值即为最佳阈值。

最大类间方差法以式(3):

$F(k) = [\mu\omega(k) - \mu(k)] \cdot 2 / [\omega(k)(1 - \omega(k))]$  (3) 作为准则函数。

式中  $\mu = \sum_{i=0}^{L-1} iP_i$ ,  $\mu(k) = \sum_{i=0}^k iP_i$ ,  $\omega(k) = \sum_{i=0}^k P_i$ , 而最佳阈值为  $K = \text{Arg} \min_{k \in \{0, 1, 2, \dots, L-1\}} F(k)$ 。

式中:  $L$  为图像的灰度级,  $P_i = n_i/N$  是各灰度值的概率,  $n_i$  为图像中灰度为  $i$  的像素数,  $N = \sum_{i=0}^{L-1} n_i$  是图像中的总像素数,  $\mu$  是整幅图像的平均值,  $\mu(k)$  是阈值为  $k$  时灰度的平均值,  $\omega(k)$  是图像灰度值在  $0 \sim k$  范围内的概率。

## 2.2 采用 SAPSO 算法优化最大类间方差分割法的具体实现

从上面讨论可以看出,最大类间方差法的计算量很大,严重影响了阈值的选取效率,难以满足图像分割的实时处理要求和大幅图像处理的需要。由于最大类间方差法实际上也是一种寻优方法,所以文中使用前文提出的 SAPSO 算法优化最大类间方差法的寻优过程。

### 2.2.1 SAPSO 算法流程

下面是 SAPSO 算法的基本步骤:

1) 初始化:设定微粒群规模  $M$ , 每个微粒代表  $(0 \sim L-1)$  区间内的某个灰度值,其中  $L$  为需要处理图像的灰度级。随机对微粒群各微粒的初始位置和速度进行初始设定,预先设定算法的运行参数。

2) 根据式(1)和式(2),由当前位置和速度产生各个微粒的新的位置。并且计算微粒群中每个微粒的当前最好位置  $pbest$  和全局最好位置  $gbest$ 。

While(不满足迭代终止条件)do

3) 选择式(3)的  $F(k)$  作为微粒群算法的适应函数,计算每个粒子新位置的适应值。

4) 对各个微粒,将其适应值与所经历过的最好位置  $pbest$  的适应值进行比较,若较好,则将其作为当前的最好位置;根据各个微粒的个体极值  $pbest$  找到全局最好位置  $gbest$ 。对适应值进行比较,若较好,则将其作为当前的全局最好位置。进行一步迭代。

5) 根据式(1)进化每个微粒的  $V_i$ ,并把它限制在  $v_{max}$  内。

6) 根据式(2)进化每个微粒的  $X_i$ ,并把它限制在  $0 \sim L-1$  内。

7) 计算两个位置所引起的适应值的变化量  $\Delta E$ ; 若  $\Delta E > -e$ ,  $e$  为允许目标函数变坏范围,接受新值;否则就拒绝,  $x_{k+1}$  仍为  $x_k$ 。

End

8) 将  $gbest$  对应的阈值  $K$  作为分割阈值门限,进行图像分割。利用灰度图像直方图可以实现灰度图像的二值化,一般而言,如果一幅图像用  $f(i, j)$  表示,图像的二值化阈值处理方式如下:

$$f(i, j) = 1; f(i, j) \geq K$$

$$f(i, j) = 0; f(i, j) < K$$

通常,用  $f(i, j) = 1$  的部分表示图像,用  $f(i, j) = 0$  的部分表示背景<sup>[13]</sup>。

### 2.2.2 SAPSO 算法的主要参数的选择

虽然粒子群算法发展迅速并取得了可观的研究成果,但其理论基础仍相对薄弱,尤其是算法基本模型中的参数设置和优化问题还缺乏成熟的理论论证和研究。目前比较通用的参数设置方法是经验法和实验法。

粒子群规模的选择, A EI-Gallad 在其研究论文中利用统计方法分析了微粒群规模、速度极限和迭代次数对算法性能的影响<sup>[14]</sup>,但是这三个参数与具体问题的特性密切相关,文中算法经过实验得到粒子数为 5 比较合适。

惯性权重  $w$ 、加速常数  $c_1$ 、 $c_2$  和最大速度  $v_{max}$  共同维护微粒对全局和局部搜索能力的平衡。这四个算法参数的设置与具体问题密切相关,目前比较常用的方法是针对具体求解问题,利用充分的实验来确定<sup>[15]</sup>。

加速常数  $c_1$ 、 $c_2$ , 低的值允许微粒在被拉回之前可以在目标区域外徘徊,而高的值则导致微粒突然地冲向或越过目标区域,一般都取 2,不过在文献中也有其它的取值,但一般  $c_1 = c_2$  并且范围在  $0 \sim 4$  之间<sup>[11]</sup>, J Kennedy 和 R Eberhart 曾在论文中建议设置  $c_1 = c_2 = 2$  以保证式(1)中的随机乘积均值等于 1,文中算法中统一取  $c_1 = c_2 = 2$ 。

惯性权重因子  $w$  使微粒保持运动惯性,使微粒有扩展搜索空间的趋势,有能力探索新的区域。在此取值与经典粒子群算法一致,  $w = 1$ 。

速度限制最大值  $v_{max}$  决定当前位置与最好位置之间的区域的分辨率(或精度)。如果  $v_{max}$  太高,微粒可能会飞过好解;如果  $v_{max}$  太小,微粒不能在局部好区间之外进行足够的探索,导致陷入局部最优。

该限制有 3 个目的:





图 3 图像 b 分割效果比较

3.2 计算效率

文中在 Matlab 7.1 环境下进行仿真实验,运行参数设置如下:粒子群规模  $M = 5$ ,加速常数  $c_1 = c_2 = 2$ ,惯性权重因子  $w = 1$ ,速度限制最大值  $v_{\max} = 4$ ,位置限制最小值  $x_{\min} = 0$ ,位置限制最大值  $x_{\max} = L - 1$ ,在实验中采用常见的 256 灰度级的图像,故  $L$  为 256,退火系数  $\alpha = 0.99$ ,允许目标变坏的范围  $e = 80$ 。

首先,利用传统最大类间方差法,求出最佳阈值所对应的最大类间方差,求得图 a 的最佳阈值为 118,对应最大类间方差为 1588,图 b 最佳阈值为 74,对应最大类间方差为 1138。用文中提出的各种算法记录“达到要求”的迭代次数,这里达到要求意味着目标函数到达最佳阈值对应的最大类间方差,各种算法各随机测试 100 次,结果如表 3、表 4 所示。

表 3 a 图像各种算法的测试结果

算法	最快迭代次数	最慢迭代次数	平均迭代次数
基本 PSO 算法	2	29	13.36
方法 1 算法	1	29	11.75
方法 2 算法	1	64	15.18
方法 3 算法	1	26	10.84
SAPSO 算法	1	20	9.5

表 4 b 图像各种算法的测试结果

算法	最快迭代次数	最慢迭代次数	平均迭代次数
基本 PSO 算法	1	35	10.97
方法 1 算法	1	42	10.37
方法 2 算法	1	45	13.65
方法 3 算法	1	26	9.55
SAPSO 算法	1	19	8.5

方法 1 比基本粒子群算法有所改进。方法 2 不如方法 3 和方法 4。因为方法 2 中,随着温度的降低,接受的概率变得很小,可能使有些粒子的位置停滞不前,反而使收敛很慢。从表 3 与表 4 可以看出,SAPSO 算法是一种比较有效的算法,它以最少的平均迭代次数求得了图像的最佳分割阈值,实时性更强。

从运算时间来看,最大类间方差的图像分割算法主要包括了方差计算与图像阈值分割运算,对于每一

种分割算法,图像的阈值分割运算时间是一样的,因此其关键是减少方差运算的时间,传统方法求解最优的阈值  $K$ ,必须对每一个灰度值  $k$  计算  $\mu, \mu(k), \omega(k)$  3 个量,然后按照公式求出方差,当所有灰度  $k$  值计算完成后进行搜索,找到最大的方差及其对应的灰度值  $K$ ,以  $K$  作为最佳阈值分割图像,设每一个灰度值计算方差的时间为  $t$ ,则总的计算时间为  $256t$ 。

而文中采用的 SAPSO 算法比最大类间方差法计算时间少得多,SAPSO 算法中,对于平均代数不到 10,粒子数为 5 的组合,总的平均计算时间不到  $50t$ 。显然,文中算法在得到和传统最大类间方差算法相近的视觉效果的同时,速度比传统算法有了明显的提高。

4 结束语

文中采用了基于模拟退火思想粒子群算法的最大类间方差图像分割方法,该方法充分发挥了粒子群算法、模拟退火算法、最大类间方差法三种算法的优势,非线性快速且稳定地求解最大类间方差及对应的灰度阈值,在取得较好的图像分割效果的同时,显著地提高了计算速度。仿真实验结果表明,其计算速度远胜于普通最大类间方差图像分割方法,不仅极大地提高了搜索效率又保证了图像分割的效果。权衡分割精度和计算效率两个方面,文中方法不失为一种实用有效的图像分割算法。

参考文献:

- [1] 梁光明,刘东华.用于显微细胞图像的二维自适应阈值分割算法的优化[J].中国图象图形学报,2003,8(7):764-768.
- [2] 王秋光.基于图像边缘信息的 2 维阈值分割方法[J].中国图象图形学报,2007,12(1):78-81.
- [3] 于立强.基于 PSO 的图像分割方法与 GPU 加速的蚁群算法[D].大连:大连理工大学,2007.
- [4] 苏彩虹,吴菁,朱学峰.改进的粒子群算法用于图像分割[J].佛山科学技术学院学报,2007,25(3):15-18.
- [5] 朱小六,熊伟丽,徐保国.基于模拟退火技术的 QDPSO 算法[J].计算机工程,2007,33(15):209-210.
- [6] Kennedy J, Eberhart R C. Particle Swarm Optimization[C]//IEEE International Conference on Neural Networks. Piscataway, NJ: IEEE Press, 1995:1942-1948.
- [7] Eberhart R C, Kennedy J. A New Optimizer Using Particle Swarm Theory[C]//Proc of the 6th International Symposium on Micro Machine and Human Science. Nagoya, Japan: [s. n.], 1995:9-43.
- [8] 高尚,杨静宇.群智能算法及其应用[M].北京:中国水利水电出版社,2006.

表 1 粒子迭代结果

迭代次数	粒子 1	粒子 2	粒子 3	粒子 4	粒子 5	粒子 6	粒子 7	粒子 8	粒子 9	粒子 10	全局最优解	执行时间(h)
1	23145	25143	35412	52134	25143	32541	34125	43251	41325	45231	32541	12.9
2	21453	41325	34251	54321	31245	23415	14325	21543	45231	13425	41325	13
3	15324	13524	15423	35241	23154	32541	32154	41523	13524	14352	32541	12.9
4	43152	35412	54123	34251	32415	23415	21453	53412	25413	15324	35421	13
5	41523	23514	15324	14325	13524	31542	42135	35142	52431	42315	23514	12.9
6	43215	15324	45213	15432	32415	23541	15432	53214	25431	15234	53214	12.9
7	32541	32451	32154	51243	35412	14532	14253	54321	53412	32541	32541	12.9
8	23514	14523	15234	34251	43152	23541	53241	12543	14253	21543	23514	12.9
9	32514	43125	23154	12453	13542	34521	41352	32451	53421	13524	32512	12.9
10	23514	24513	13254	24153	14523	24531	23541	52413	25413	21534	23541	12.9

表 2 实验结果

	最优解	最差解	平均解
处理时间(单位 h)	12.9	14.1	13.29

选取第十次迭代获得最优调度序列  $< wf_2, wf_3, wf_5, wf_4, wf_1 >$  为最终的调度方案,资源如下:

Relation = { ( task<sub>11</sub>, reobj<sub>12</sub> ), ( task<sub>12</sub>, reobj<sub>22</sub> ), ( task<sub>13</sub>, reobj<sub>32</sub> ), ( task<sub>14</sub>, reobj<sub>42</sub> ), ( task<sub>21</sub>, reobj<sub>11</sub> ), ( task<sub>22</sub>, reobj<sub>21</sub> ), ( task<sub>23</sub>, reobj<sub>31</sub> ), ( task<sub>24</sub>, reobj<sub>41</sub> ), ( task<sub>31</sub>, reobj<sub>12</sub> ), ( task<sub>32</sub>, reobj<sub>21</sub> ), ( task<sub>33</sub>, reobj<sub>31</sub> ), ( task<sub>34</sub>, reobj<sub>42</sub> ), ( task<sub>41</sub>, reobj<sub>13</sub> ), ( task<sub>42</sub>, reobj<sub>21</sub> ), ( task<sub>43</sub>, reobj<sub>31</sub> ), ( task<sub>44</sub>, reobj<sub>41</sub> ), ( task<sub>51</sub>, reobj<sub>13</sub> ), ( task<sub>52</sub>, reobj<sub>22</sub> ), ( task<sub>53</sub>, reobj<sub>22</sub> ), ( task<sub>54</sub>, reobj<sub>41</sub> ) }。

4 结束语

离散粒子群算法具迭代速度快,易于实现的优点,工作流任务的调度是工作流研究的重点内容。文中将二者相结合,给出了工作流的定义,将离散粒子群算法应用工作流任务调度的过程中,并通过实验证明了应用的有效性及其合理性。

参考文献:

[1] Georgakopoulos D, Homick M. An Overview of Workflow

Management: From Process Modeling to Workflow Automation Infrastructure [J]. Distribute and Parallel Databases, 1995(3):119-153.

[2] Workflow Management Coalition. Workflow management coalition terminology and glossary[R]. Brussels: Workflow Management Coalition, 1996.

[3] 吴立峰,金 烨. 工作流系统的资源管理问题[J]. 计算机工程,2007(10):58-63.

[4] 肖郑进,何钦铭,陈 奇. 模糊环境中工作流任务分配的多级模型[J]. 计算机研究与发展,2007,44(2):302-309.

[5] 刘丹妮,陈秀寓. 一种基于工作流图的时间资源分配策略[J]. 计算机应用,2006,26(9):2166-2171.

[6] Cagnina L, Esquivel S, Gallard I R. Particle Swarm Optimization for Sequencing Problems: A Case Study[C]// In: Proc of the Congress on Evolutionary Computation. Oregon, Portland:[s. n. ], 2004:536-541.

[7] 周 燕,姜 浩. 面向动态工作流的 Web 服务组合模型研究[J]. 计算机技术与发展,2008,18(1):63-66.

[8] 宋佩华,蒋联源,欧启忠. 基于离散粒子群优化算法求解矩形件排样问题[J]. 计算机应用与软件,2008,25(1):238-240.

[9] 乐晓波,李京京,唐贤瑛. 基于 Petri net 建模的资源调度的蚁群算法[J]. 计算机技术与发展,2006,16(1):44-46.

(上接第 87 页)

[9] Poli R, Kennedy J, Blackwell T. Particle swarm optimization [J]. Swarm Intell,2007(1):33-57.

[10] 黄辉先,陈资滨. 一种改进的粒子群优化算法[J]. 系统仿真学报,2007,19(21):4922-4925.

[11] 吕振肃,侯志荣. 自适应变异的粒子群优化算法[J]. 电子学报,2004,32(3):416-420.

[12] Marques L, Nunes U, de Almeida A T. Particle swarm-based olfactory guided search[J]. Auto Robot,2006,20:277-287.

[13] 陆宗骥,金登南. Visual C++ .NET 图像处理编程[M]. 北

京:清华大学出版社,2006.

[14] El-Gallad A, El-Hawary M, Sallam A, et al. Enhancing the particle swarm optimizer via proper parameters selection[C]// IEEE CCECE02 Proceedings. Canadian: IEEE Press, 2002: 792-797.

[15] 彭 宇,彭喜元,刘兆庆. 微粒群算法参数效能的统计分析[J]. 电子学报,2004,32(2):209-213.

[16] 王 凌,刘 波. 微粒群优化与调度算法[M]. 北京:清华大学出版社,2008:45-50.